

Lecture 9: Method Overloading and Operator Overloading

1. Introduction

Overloading allows methods or operators to perform different tasks based on their input parameters or operands. This increases code readability and flexibility.

2. Method Overloading in Python

Python does **not** support method overloading directly by defining multiple methods with the same name but different parameters. Instead, it can be achieved by default arguments or variable arguments.

Example with default arguments:

```
class Math:
    def add(self, a, b=0, c=0):
        return a + b + c

m = Math()
print(m.add(5))           # Output: 5
print(m.add(5, 10))      # Output: 15
print(m.add(5, 10, 15)) # Output: 30
```

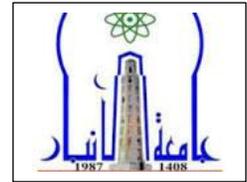
3. Operator Overloading

Operator overloading allows us to define the behavior of operators for user-defined classes by implementing special methods.

Example: Overloading the + operator.

```
class Point:
    def __init__(self, x, y):
        self.x = x
```

Name: Hamsa M Ahmed
Subject: OOP Practical
Department: Computer Network System



```
        self.y = y

    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)

    def __str__(self):
        return f"({self.x}, {self.y})"

p1 = Point(1, 2)
p2 = Point(3, 4)
print(p1 + p2)  # Output: (4, 6)
```

4. Common Operator Overloads

Operator	Method	Description
+	<code>__add__</code>	Addition
-	<code>__sub__</code>	Subtraction
*	<code>__mul__</code>	Multiplication
/	<code>__truediv__</code>	Division
==	<code>__eq__</code>	Equality comparison
<	<code>__lt__</code>	Less than

5. Exercises

1. Create a `Vector` class with `x`, `y`, and `z` coordinates and overload the `+` and `-` operators.
2. Implement the `__str__` method to display vector coordinates nicely.
3. Test the class by adding and subtracting vector instances.

6. Summary

While Python does not support traditional method overloading, operator overloading is a powerful tool to enhance user-defined classes and make them more intuitive.

7. References

1. Python Docs - Data Model
2. Real Python - Operator Overloading